REMARKS

Applicants respectfully request that the above-identified application be re-examined.

A final Office Action in this application mailed August 4, 2006 ("Office Action"), rejected Claims 1-54. The Office Action continued the rejection of Claims 1-54 under 35 U.S.C. § 102(e) as being anticipated by Wason et al., U.S. Patent No. 6,701,383. Even though applicants believe that the claims are allowable as presented, in order to advance the prosecution of this application, Claims 1-6, 8, 12, 17-34, 39-47, 49-51, and 53-54 have been amended to further distinguish the claimed subject matter from the cited reference. Applicants respectfully submit that all pending claims in the present application are allowable for the reasons set forth below.

Prior to discussing in detail why applicants believe that all of the claims in this application are allowable, a brief description of the disclosed subject matter and a brief description of the teachings of the cited and applied reference are provided. The following discussions of the disclosed subject matter and the cited and applied reference are not provided to define the scope or interpretation of any of the claims of this application. Instead, these discussions are provided to help the U.S. Patent and Trademark Office better appreciate important claim distinctions discussed thereafter.

Disclosed Subject Matter

A system for synchronizing the playback of media content with other content or with host computer time information is disclosed. The system includes a web browser, a plurality of media players, and a player-hosting peer. The web browser provides timing information to each of the plurality of media players. Each of the media players includes a first interface for object management and a second interface for exchanging timing and synchronization with the web browser. *The player-hosting peer is located within the web browser. The player-hosting peer*

-11-

negotiates a playback state and a rendering status between the web browser and each of the plurality of media players by exchanging, without user input, command and state change information between the web browser and the media players.

Also disclosed is a method of synchronizing the playback of media content with other content or with host computer time information. The method comprises providing a timing representation to each of a plurality of media players. The method also comprises providing a first media player interface for object management and a second media player interface for exchanging timing and synchronization information with a web browser. *The method further comprises issuing commands from the web browser to each of the plurality of media players, the commands being directed to media player operations other than, and in addition to, instantiation of the plurality of media players; and notifying the web browser of changes of the states of the plurality of media players.*

Wason et al., U.S. Patent No. 6,701,383 B1 ("Wason")

Wason is generally directed towards an extensible software framework with an overlaying abstraction layer providing a cross-platform interface (abstract). More specifically, Wason discloses a synchronization abstraction layer ("SAL") providing a uniform interface between the framework and one or more plug-ins (Col. 2, lines 26-27). The SAL synchronizes itself and other plug-ins to a timeline provided by the underlying software framework. The plug-ins interact with the underlying framework through the SAL (Col. 2, lines 30-35). Plug-ins are software extension modules that are dynamically loaded into computer memory at run-time (Col. 1, lines 57-60). SAL is an extension module (Col. 3, lines 61-62). Wason et al. discloses a content framework that insulates content and plug-in developers from details and differences in hardware platforms, so that the same content or plug-in can run on different desktop platforms, for example, PC, Macintosh, and Linux (Col. 2, lines 51-55).

MSFT\21165AM4.DOC

Wason et al. does not disclose or suggest a player-hosting peer located within a web browser that negotiates a playback state and a rendering status between the web browser and a number of media players by exchanging command and state change information between the web browser and the media players without user input. Nor does Wason et al. disclose a method that comprises issuing commands from a web browser to a number of media players, the commands being directed to media player operations other than, and in addition to, instantiation of the media players; and notifying the web browser of the media players' state changes.

Rejection of Claims 1-54 Under 35 U.S.C. § 102(e)

As noted above, Claims 1-54 were rejected under 35 U.S.C. § 102(e) by the Office Action "as being anticipated by Wason." Applicants respectfully disagree. Independent Claim 1 recites, in its entirety:

> 1. A system for synchronizing the playback of media content with other content or with host computer time information, the system comprising:
>
> a web browser for providing a timing representation to each of a plurality of media players;
>
> a plurality of media players, each of the plurality of media players including a first interface for object management and a second interface for exchanging timing and synchronization information with the web browser; and
>
> a player-hosting peer within the web browser for negotiating a playback state and a rendering status between the web browser and *each of the plurality of media players by exchanging, without user input, command and state change information between the web browser and each of the plurality of media players.*

(Emphasis added.)

-13-

Wason et al. does not teach or suggest a *player-hosting peer* within the Web browser for *negotiating*[1] *a playback state and a rendering status between the Web browser and each of a plurality of the media players*. Wason et al. discloses a synchronization abstraction layer (SAL) that provides synchronization between plug-ins and an underlying framework. (Emphasis added; Col. 2, lines 31-33.) The SAL 310 provides a software interface through which a media player, such as RealPlayer 301, can communicate with other extension modules such as RealTOC 314 for exchange of data. (Col. 4, lines 56-60; Col. 5, lines 11-20.) Figure 3 of Wason et al. does not teach or suggest a player-hosting peer for *negotiating* a playback state and a rendering status *between the Web browser and each of the plurality of the media players, by exchanging command and state change information* as recited in Claim 1. Wason et al. discloses a SAL 310 interface between the RealPlayer 301 and other extension modules such as RealTOC 314. Wason et al. does not disclose a player-hosting peer that actively manages a number of media players by negotiating playback state and rendering status between a Web browser and each of the media players by issuing commands and providing status. The SAL 310 interface disclosed by Wason et al. provides interface and data format compatibility and a channel for passing data between two incompatible entities on the different sides of the interface. When the SAL 310 functions as an interface between the RealPlayer 301 and RealTOC 314, the RealPlayer 301 periodically calls SAL 310 with the current time and SAL 310 passes the current time information to JVM 313, which, in turn, notifies RealTOC 314. (See Col. 5, lines 57-61.) Nowhere does the SAL 310 interface *negotiate a playback state and a rendering status*. All

---

[1] Those skilled in the art will appreciate that "negotiating" is generally understood to mean the exchange of command and status information between the negotiating parties (for example, see Andrew S. Tanenbaum, *Computer Networks*, 25, 375, 2d ed., Prentice Hall 1988; J.C. Macklin & Ian McLean, *Windows Server 2003 Network Infrastructure*, 11-27, Microsoft Press 2004). Claim 1 has been amended so as to clarify this understood meaning.

MSFT\21165AM4.DOC

SAL 310 interface does is pass time information between RealPlayer 301 and RealTOC 314. Those skilled in the art will appreciate that SAL is a type of software middle layer that is generally used to couple an upper layer software module with a number of incompatible lower layer software modules. The middle layer accomplishes this coupling by defining standard interfaces on the lower and upper sides and requiring the corresponding software modules to comply with such interfaces. SAL performs this middle layer function for different platforms, wherein the same plug-in may be used on the different platforms. As noted above, Wason et al. discloses an abstraction layer (SAL) that provides a _cross-platform interface_ for plug-ins (Col. 2, lines 42-44). This is in contrast to having a _plurality of media players_ running simultaneously on the _same platform_, i.e., a web browser. Therefore, Claim 1 is submitted to be allowable for at least the reasons presented above.

Claims 2-38 depend from Claim 1 and are submitted to be allowable for at least the same reasons presented above with respect to Claim 1.

Independent Claim 39 recites, in its entirety:

> 39. A method of synchronizing the playback of media content with other content or with host computer time information, the method comprising the steps of:
>
> providing a timing representation to each of a plurality of media players;
>
> providing a first media player interface for object management and a second media player interface for exchanging timing and synchronization information with a web browser;
>
> _issuing commands from the web browser to each of the plurality of media players, the commands being directed to media player operations other than, and in addition to, instantiation of the plurality of media players; and_
>
> _notifying the web browser of changes of the states of the plurality of media players._

MSFT\21165AM4.DOC

(Emphasis added.)

Wason et al. does not teach or suggest *notifying* the Web browser of each of a plurality of media players' *state changes*. Wason et al. discloses the SAL 310 that passes current time information between RealPlayer 301 and other extension modules such as RealTOC 314. (Col. 5, lines 57-61.) Those skilled in the art appreciate that state changes take place in response to a particular event such as a command or other software event. Continuous passage of time and notification of current time do not constitute events for the purpose of state changes, unless a particular point in time is defined as an event for a state change. Wason et al. does not teach or suggest any such event associated with the passage of time. Therefore, Wason et al. does not teach or suggest *notifying the web browser of state changes of each of plurality of the media players*. Therefore, independent Claim 31 is submitted to be allowable for at least the reasons presented above.

Claims 40-54 depend from Claim 39 and are submitted to be allowable for at least the same reasons discussed above with respect to Claim 39.

CONCLUSION

In summary, applicants respectfully submit that all the claims in this application are clearly allowable in view of the disclosures of Wason et al. As a result, applicants respectfully request that all of the claims remaining in this application be allowed and that this application be

MSFT\21165AM4.DOC

passed to issue. If the Examiner has any questions, the Examiner is invited to contact applicants' attorney at the number set forth below.

Respectfully submitted,

CHRISTENSEN O'CONNOR
JOHNSON KINDNESS PLLC

Gary S. Kindness
Registration No. 22,178
Direct Dial No. 206.695.1702

I hereby certify that this correspondence is being deposited with the U.S. Postal Service in a sealed envelope as first-class mail with postage thereon fully prepaid and addressed to Mail Stop AF, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on the below date.

Date: _Nov. 3, 2006_   _Patricia Seebeck_

FXM:aew

MSFT\21165AM4.DOC